

## Program stat3d.c

```
*****  
/*  
 * Program: stat3d.c  
 * Purpose: To read in a 3-D image and output phase volumes  
 * and report the volume and pore-exposed surface area  
 * fractions  
 * Programmer: Dale P. Bentz  
 * NIST  
 * 100 Bureau Drive Mail Stop 8615  
 * Gaithersburg, MD 20899-8615  
 * Phone: (301) 975-5865  
 * E-mail: dale.bentz@nist.gov  
 */  
*****  
/* This software was developed at the National Institute of */  
/* Standards and Technology by employees of the Federal Government */  
/* in the course of their official duties. Pursuant to title 17 */  
/* Section 105 of the United States Code this software is not */  
/* subject to copyright protection and is in the public domain. */  
/* CEMHYD3D is an experimental system. NIST assumes no */  
/* responsibility whatsoever for its use by other parties, and */  
/* makes no guarantees, expressed or implied, about its quality, */  
/* reliability, or any other characteristic. We would appreciate */  
/* acknowledgement if the software is used. This software can be */  
/* redistributed and/or modified freely provided that any */  
/* derivative works bear some notice that they are derived from it, */  
/* and any modified versions bear some notice that they have been */  
/* modified. */  
  
#include <stdio.h>  
#include <math.h>  
  
#define ISIZE 100  
#define NPHASES 50  
  
main(){  
    static int mic [ISIZE] [ISIZE] [ISIZE];  
    int valin,ix,iy,iz;  
    int ix1,iy1,iz1,k;  
    long int voltot,surftot,volume[NPHASES],surface [NPHASES];  
    FILE *infile,*statfile;  
    char filen[80],fileout[80];  
  
    printf("Enter name of file to open \n");  
    scanf("%s",filen);  
    printf("%s \n",filen);  
    printf("Enter name of file to write statistics to \n");  
    scanf("%s",fileout);  
    printf("%s \n",fileout);  
  
    for(ix=0;ix<=(NPHASES-1);ix++){  
        volume[ix]=surface[ix]=0;  
    }  
}
```

```

infile=fopen(filen,"r");
statfile=fopen(fileout,"w");

/* Read in image and accumulate volume totals */
for(iz=0;iz<ISIZE;iz++){
    for(iy=0;iy<ISIZE;iy++){
        for(ix=0;ix<ISIZE;ix++){
            fscanf(infile,"%d",&valin);
            mic [ix] [iy] [iz]=valin;
            if(valin<NPHASES){
                volume[valin]+=1;
            }
            else{
                volume[0]+=1;
            }
        }
    }
}
fclose(infile);

for(iz=0;iz<ISIZE;iz++){
    for(iy=0;iy<ISIZE;iy++){
        for(ix=0;ix<ISIZE;ix++){
            if((mic [ix] [iy] [iz]!=0)&&(mic[ix][iy][iz]<=49)){
                valin=mic [ix] [iy] [iz];
                /* Check six neighboring pixels for porosity */
                for(k=1;k<=6;k++){

                    switch (k){
                        case 1:
                            ix1=ix-1;
                            if(ix1<0){ix1+=ISIZE;}
                            iy1=iy;
                            iz1=iz;
                            break;
                        case 2:
                            ix1=ix+1;
                            if(ix1>=ISIZE){ix1-=ISIZE;}
                            iy1=iy;
                            iz1=iz;
                            break;
                        case 3:
                            iy1=iy-1;
                            if(iy1<0){iy1+=ISIZE;}
                            ix1=ix;
                            iz1=iz;
                            break;
                        case 4:
                            iy1=iy+1;
                            if(iy1>=ISIZE){iy1-=ISIZE;}
                            ix1=ix;
                            iz1=iz;
                            break;
                        case 5:
                            iz1=iz-1;
                            if(iz1<0){iz1+=ISIZE;}
                            iy1=iy;
                }
            }
        }
    }
}

```

```

        ix1=ix;
        break;
    case 6:
        iz1=iz+1;
        if(iz1>=ISIZE){iz1-=ISIZE;}
        iy1=iy;
        ix1=ix;
        break;
    default:
        break;
    }
}
if((ix1<0)|| (iy1<0)|| (iz1<0)|| (ix1>=ISIZE)|| (iy1>=ISIZE)|| (iz1>=ISIZE)){
    printf("%d %d %d \n",ix1,iy1,iz1);
    exit(1);
}
if((mic[ix1][iy1]
[iz1]==0)|| (mic[ix1][iy1][iz1]>44)){
    surface[valin]+=1;
}
}
}
}

printf("Phase      Volume      Surface      Volume      Surface \n");
printf(" ID      count      count      fraction   fraction \n");
fprintf(statfile,"Phase      Volume      Surface      Volume      Surface \n");
fprintf(statfile," ID      count      count      fraction   fraction \n");
/* Only include clinker phases in surface area fraction calculation */
surftot=surface[1]+surface[2]+surface[3]+surface[4];
voltot=volume[1]+volume[2]+volume[3]+volume[4];
k=0;
printf(" %d      %8ld      %8ld \n",k,volume[0],surface[0]);
fprintf(statfile," %d      %8ld      %8ld \n",k,volume[0],surface[0]);
for(k=1;k<=4;k++){
printf(" %d      %8ld      %8ld      %.5f      %.5f\n",k,volume[k],surface[k],
(float)volume[k]/(float)voltot,(float)surface[k]/(float)surftot);
    fprintf(statfile," %d      %8ld      %8ld      %.5f
%.5f\n",k,volume[k],surface[k],
(float)volume[k]/(float)voltot,(float)surface[k]/(float)surftot);

}
printf("Total      %8ld      %8ld\n\n\n",voltot,surftot);
fprintf(statfile,"Total      %8ld      %8ld\n\n\n",voltot,surftot);
for(k=5;k<=30;k++){
    printf(" %d      %8ld      %8ld\n",k,volume[k],surface[k]);
    fprintf(statfile," %d      %8ld      %8ld\n",k,volume[k],surface[k]);
}
printf(" 35      %8ld      %8ld\n",volume[35],surface[35]);
fprintf(statfile," 35      %8ld      %8ld\n",volume[35],surface[35]);
printf(" 45      %8ld      %8ld\n",volume[45],surface[45]);
fprintf(statfile," 45      %8ld      %8ld\n",volume[45],surface[45]);
fclose(statfile);
}

```